

# Java, LEDs und ein RaspberryPi: Ein Projektversuch mit Erstsemestern

Axel Langhoff, Michael Striewe, Michael Goedicke

Universität Duisburg-Essen

{axel.langhoff,michael.striewe,michael.goedicke}@paluno.uni-due.de

## Zusammenfassung

Bei der Einführung von Studierenden in die Programmierung stellt sich immer wieder das Problem, den Stoff attraktiv aufzubereiten und den Studierenden eine Möglichkeit zu geben, ihr Wissen auf eine motivierende Art anzuwenden. Für Studierende ohne Vorkenntnisse ist die Einstiegshürde in eine vollwertige Programmiersprache hoch. Hinzu kommt die Schwierigkeit mit dem frisch erworbenen Wissen eigene Ideen in ansehnliche Ergebnisse umzusetzen. Für Studierende mit großem Vorwissen hingegen besteht die realistische Gefahr sich in diesen Veranstaltungen zu langweilen.

In dem hier vorgestellten Ansatz haben wir versucht, unseren Studierenden in Form eines LED-Boards eine Umgebung zur Verfügung zu stellen, die (1) im Sinne einer Lernumgebung möglichst einfach zu verwenden ist, (2) dem unterschiedlichen und im Verlauf der Vorlesung wachsenden Kenntnisstand der Studierenden gerecht wird, (3) sichtbare und schöne Ergebnisse der Programmieranstrengungen der Studierenden ermöglicht, (4) der Kreativität der Studierenden möglichst wenig Grenzen setzt, und (5) die Motivation der Studierenden steigert. Diesen Zielen liegt die Annahme zugrunde, dass eine motivierte Beschäftigung mit einem ansprechenden Projekt den Studierenden hilft, ihre Programmierkenntnisse zu vertiefen und zu festigen.

Neben dem LED-Board selbst berichten wir in diesem Beitrag über unsere Erfahrungen mit der Verwendung des LED-Boards in unserer Programmierungsvorlesung, sowie die Akzeptanz und den Lernerfolg unter den Studierenden.

## 1 Das LED-Board

Das von uns entwickelte LED-Board besteht ausschließlich aus Komponenten, die im regulären Elektronikhandel problemlos erhältlich sind und ohne besondere handwerkliche Qualifikation in der passenden Form zusammengebaut werden können. Die beiden wesentlichen Bestandteile sind abschneidbare LED-

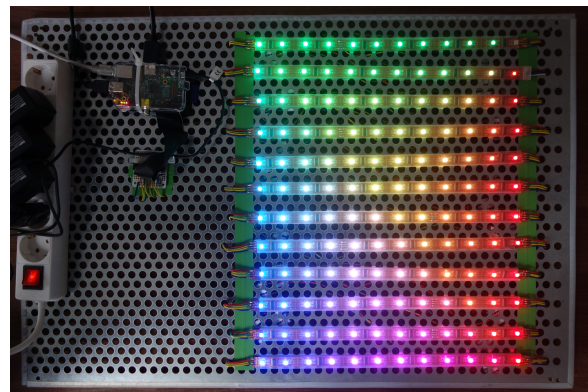


Abbildung 1: LED-Board

Bänder und ein RASPBERRYPI<sup>1</sup>, um die LEDs auf den Bändern anzusteuern. Da die LEDs auf den Bändern einen Abstand von etwa 3,5 cm haben, haben wir uns für unsere Installation für ein Feld von 12x12 LEDs entschieden, das damit eine Fläche von etwa 40 x 40 cm einnimmt. Als Träger für die Montage der LEDs dient eine handelsübliche Blech-Lochplatte aus dem Baumarkt, so dass die Kabelführung bequem auf der Rückseite des Boards erfolgen kann. Jeder LED kann unabhängig von den Farben der anderen LEDs ein individueller RGB-Farbwert zugewiesen werden, so dass sich ein großes und leuchtkräftiges 144-Pixel-Farbdisplay ergibt (siehe Abbildung 1). Nach anfänglichen technischen Schwierigkeiten beim Zusammenbau, die teilweise auf mangelnde Dokumentation der verwendeten Komponenten zurückzuführen sind, konnten wir nach dem ersten Prototypen ein zweites, baugleiches Board an einem Arbeitstag zusammenbauen.

Um die direkte Ansteuerung der LEDs über die GPIO-Pins des RASPBERRYPI vor den Studierenden zu verbergen und die Ansteuerung der LEDs möglichst einfach zu gestalten haben wir eine Java-Bibliothek implementiert. Die Wahl fiel auf Java, da auch unsere Vorlesung für die das Board entwickelt wurde auf Ja-

<sup>1</sup><http://www.raspberrypi.org/>

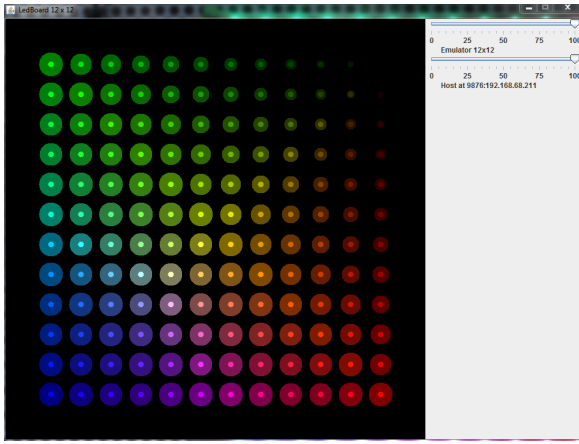


Abbildung 2: Emulator zum LED-Board. Die Größe der Kreise korrespondiert mit der Helligkeit der LEDs auf dem physischen Board. Mit den Schiebereglern rechts kann die Helligkeit des Emulators und des angeschlossenen Boards kontrolliert werden.

va aufbaut. Diese Bibliothek bietet mehrere leicht verständliche Methoden, um die einzelnen LEDs über ihre Koordinaten auf dem Board anzusteuern. Außerdem ermöglicht die Bibliothek den Studierenden am heimischen PC Programme für das Board zu entwickeln und die Ergebnisse unabhängig vom Board in einem Emulator zu sehen und zu testen (siehe Abbildung 2). Neben der Steuerung und Emulation des Boards wird von der Bibliothek auch ein einfaches Interface zur Verfügung gestellt, um Tastatureingaben auszuwerten und so die Entwicklung interaktiver Programme zu ermöglichen. Dieses Interface, wie auch der Rest der Bibliothek, verbergen außerdem das notwendige Threadhandling vollständig vor dem Benutzer.

Die Bibliothek stellt keine weiteren Anforderungen an die Umgebung und kann somit ohne Umstände durch Eintragung in den Java-Classpath in ein neues Java-Projekt eingebunden werden. Zu der Bibliothek wurden eine kleine, an Anfänger gerichtete Anleitung sowie mehrere Beispielprogramme angeboten, um die Verwendung der Bibliothek zu demonstrieren.

## 2 Einsatzziele des Boards in der Lehre

Ziel der Entwicklung des LED-Boards war es, eine Lernumgebung zu schaffen, die insbesondere in Lernveranstaltungen mit Erstsemestern eingesetzt werden kann. Das bedeutet, dass sowohl Studierende ohne Vorkenntnisse, als auch Studierende mit Programmiererfahrung aus Schule oder Freizeit angesprochen werden sollten, und dass das Board mit dem fortschreitenden Kenntnisstand im Laufe der Vorlesung mithalten sollte. Insbesondere war angestrebt, mit dem Board eine freie Umgebung anzubieten, in der die Studierenden keine vorgegebenen Aufgaben zu lösen haben,

sondern eigene Projekte realisieren können. Auf diese Weise erhofften wir drei Ziele zu erreichen:

- Steigerung der Motivation der Studierenden aufgrund von eigener kreativer Tätigkeit,
- Vertiefung der vorhandenen Programmierkenntnisse durch praktische Anwendung,
- Sensibilisierung für Themen, die über den Stoff der Vorlesung bzw. die vorhandenen Kenntnisse hinaus gehen und die sich aus den selbstgewählten Zielsetzungen ergeben.

Der letzte Punkt ist insbesondere wichtig, da er sowohl unerfahrene als auch erfahrene Studierende anspricht: Unerfahrene Studierende können an ihren eigenen Projekten feststellen, wofür sie die Inhalte der Vorlesung benötigen, während erfahrene Studierende erkennen können, dass auch ihre Fähigkeiten noch nicht umfassend sind. Beide Gruppen werden somit zum Lernen motiviert.

### 2.1 Erforderliche und trainierte Fähigkeiten

Die einfachste Verwendung des LED-Boards besteht in der Darstellung von Standbildern. Die Farbwerte der LEDs können dazu theoretisch mit einem Befehl pro LED „per Hand“ gesetzt werden. Einzig ein sehr rudimentäres Verständnis von Arrays ist hierzu erforderlich und die Ergebnisse wären weder besonders interessant, noch lehrreich. Eine Abfolge von unzusammenhängenden Bildern wäre als weitere Verwendung denkbar, die erbrachte Programmierleistung ginge aber nicht über eine reine Fleißarbeit hinaus. Als strukturierendes Merkmal des Codes drängen sich in diesem Fall lediglich einzelne Methoden auf, die jeweils ein Einzelbild zeichnen und nacheinander aufgerufen werden. Die Inhalte einer typischen Anfängervorlesung zur objekt-orientierten Programmierung werden mit derartigen Projekte nur zum Teil abgedeckt. Eine umfassende Festigung und Vertiefung des gesamten Stoffes kann damit also nicht erreicht werden. Andererseits bietet die geringe Menge von notwendigen Kenntnissen auch schwächeren Studierenden die Chance, einen sichtbaren Erfolg zu erzielen.

Höhere Anforderungen stellen Animationen dar, die nicht aus einer Abfolge „von Hand“ gezeichneter Einzelbilder bestehen, sondern die durch eine dynamische Erstellung bzw. Veränderung der Einzelbilder erfolgen. Eine solche Animation erfordert mindestens das geschickte Zusammenspiel von Arrays und Schleifen. Als strukturierendes Merkmal des Codes bieten sich hier Klassen an, die jeweils einen Teil der Animation (z.B. ein veränderliches grafisches Objekt) realisieren und geeignete Methoden bereitstellen, um Instanzen zu manipulieren (z.B. die Position des grafischen Objekts zu ändern) und auf dem Board anzuzeigen. Projekte dieser Art decken damit bereits einen wesentlichen Teil einer typischen Anfängervorlesung

zur objekt-orientierten Programmierung ab und geben insbesondere anschauliche Beispiele für die dort vermittelten Kernkonzepte.

Mit Hinzunahme der Tastaturunterstützung sind auch interaktive Programme, insbesondere Spiele denkbar. Die Reaktion auf Benutzereingaben erhöht nochmal die Anforderung an die Programmstrukturierung, vor allem wenn es sich um kontinuierlich entgegengenommene Eingaben handelt bzw. das Programm im Hintergrund unabhängig von einer Eingabe fortschreiten soll. Da die Studierenden zur Verarbeitung von Benutzereingaben auch Keyevents aus den `awt`-Bibliotheken von Java verarbeiten müssen, entsteht auch ein erster leichter Anknüpfungspunkt zur selbstständigen Arbeit mit API-Dokumentationen. Insbesondere gehen solche Projekte zumindest punktuell deutlich über den Umfang einer typischen Anfängervorlesung zur objekt-orientierten Programmierung hinaus. Studierende, die solche Projekte gestalten zeigen damit eine deutliche Motivation, mehr als nur den notwendigen Stoff zum Bestehen der Klausur zu lernen.

## 2.2 Abgrenzung zu anderen (Lern-)umgebungen

Von Lernumgebungen wie Greenfoot (Henriksen u. Kölling, 2004) unterscheidet sich das LED-Board dadurch, dass es mit seiner geringen Zahl an farbigen LEDs zwar grundsätzlich weniger „hübsche“ Möglichkeiten der Ausgabe bietet, den Nutzern insgesamt jedoch mehr Freiheiten in der Anwendung bietet. Insbesondere macht es keine Vorgaben über ein Welt-Modell wie beispielsweise `Kara`<sup>2</sup> oder das `JavaHamster-Modell`<sup>3</sup> und lässt damit auch anderen kreativen Ideen größeren Raum. Die bedeutendsten limitierenden Faktoren sind die Leistungsfähigkeit des `RASPBERRYPI`, die bei schnellen Bildwechseln zu Ruckeln führen kann, sowie die geringe Zahl an LEDs, die keine detaillierten Bilder erlaubt. Letzteres kann jedoch auch als Vorteil angesehen werden, da sich die Studierenden so auf algorithmisch und strukturell anspruchsvolle Projekte konzentrieren können, und weniger Mühe in optische Kosmetik investieren.

Gegenüber der reinen Programmierung mit Ausgabe auf der Kommandozeile (Java, C++, ...) sowie der Verwendung GUI-orientierter Sprachen wie `VisualBasic` oder `Delphi` bietet das LED-Board den Vorteil, dass sich schnell vergleichsweise spektakuläre Effekte erzielen lassen, ohne sich dazu in APIs einzuarbeiten, die den Rahmen einer Einführungsvorlesung üblicherweise sprengen bzw. Zeit für andere grundlegende Themen kosten. Dabei kommt auch zum Tragen, dass ein physisches Board mit leuchtkräftigen LEDs im Hörsaal bzw. Übungsraum einen anderen optischen Eindruck erzeugt als die Beamerprojektion einer Bild-

<sup>2</sup><http://www.swisseduc.ch/informatik/karatojava/greenfootkara/index.html>

<sup>3</sup><http://www.java-hamster-modell.de/index2.html>

schirmausgabe. In wie weit sich dieser Effekt abnutzt, konnte im Rahmen unserer bisherigen Erfahrungen nicht untersucht werden.

Als Nachteil des LED-Boards muss angesehen werden, dass es mit Kosten für die Hardware verbunden ist und nicht in beliebiger Stückzahl vorgehalten werden kann. Andererseits ist dies aufgrund des verfügbaren Emulators auch nicht notwendig.

## 3 Einbettung in unsere Lehrveranstaltung

Das LED-Board wurde bisher im Wintersemester 2013/14 und im Sommersemester 2014 jeweils in der Erstsemestervorlesung zur Programmierung an der Universität Duisburg-Essen am Campus Essen eingesetzt. Der allgemeine Aufbau der Vorlesung sieht vor, dass die Studierenden neben 4 SWS Vorlesung und 2 SWS Globalübung im Semesterverlauf sechs Testate ablegen, in denen Programmieraufgaben zu bearbeiten sind und bei denen mindestens 400 von maximal möglichen 600 Punkten erworben werden müssen, um die Zulassung zur abschließenden Klausur zu erwerben. Das LED-Board wurde in diesem Konzept ergänzt, indem die Studierenden durch die erfolgreiche Bewältigung eines Programmierprojekts mit dem Board ebenfalls 100 Punkte erwerben konnten. Ein Projekt konnte dabei sowohl von einzelnen Studierenden, als auch von einer Gruppe aus bis zu drei Studierenden bearbeitet werden. Zusätzlich zu dieser formalen Einbindung des LED-Boards in die Vorlesung wurde ein „Publikumspreis“ ausgelobt, indem an einem Vorlesungstermin am Ende des Semesters alle teilnehmenden Studierenden ihr Projekt vorstellen und mittels Applaus durch ihre Kommilitonen bewerten lassen konnten. Für den ersten Platz wurde dann ein Sachpreis vergeben.

Die erfolgreiche Projektteilnahme im formalen Sinne schloss neben der Teilnahme an dieser Präsentation zur Demonstration eines lauffähigen Programms zudem die Teilnahme an einem anschließenden Codereview ein. Dieses Codereview sollte einerseits sicherstellen, dass die Studierenden ihren Code wirklich selbst erarbeitet hatten und bot andererseits eine Möglichkeit für die Lehrenden, den Teilnehmern Feedback zu ihrem Code und der Struktur ihrer Programme zu geben.

Die oben beschriebenen Teilnahmebedingungen, das LED-Board sowie die Java-Bibliothek wurden jeweils gegen Mitte des Semesters in der Vorlesung vorgestellt und den Studierenden zur Verfügung gestellt. Zu diesem Zeitpunkt konnten wir ein Minimalverständnis für die Programmierung und die in unserer Veranstaltung verwendete Entwicklungsumgebung `Eclipse` voraussetzen. Gleichzeitig ergab sich dadurch eine Bearbeitungszeit für das Projekt von etwa sechs Wochen, wobei die Studierenden diesen Zeitrahmen durch einen späteren Start oder eine frühere Abgabe selbstständig verringern konnten.

Um einen hinreichenden Anspruch der Projekte zu gewährleisten, mussten alle Projekt-Ideen vorab mit dem Projektbetreuer abgesprochen und von diesem genehmigt werden. Die Schwelle ausreichender Komplexität wurde allerdings bewusst sehr niedrig gehalten, um auch die schwächeren Studierenden ansprechen zu können. Einzige feste Vorgabe in beiden Semestern war daher, dass ein bewegtes Bild auf dem LED-Board erzeugt werden musste. Aufgrund der Erfahrungen aus der ersten Durchführung haben wir uns beim zweiten Mal zusätzlich vorbehalten Projekte abzulehnen, die in dieser oder ähnlichen Formen schon von anderen Studierenden angemeldet worden waren, da sonst die Vorstellung am Ende des Semesters unter zu vielen ähnlichen Projekten gelitten hätte. Wurden Vorschläge aufgrund dieser Kriterien abgelehnt, wurden vom Projektbetreuer in der Regel Anregungen gegeben, wie die Projektidee geeignet abgewandelt werden konnte, um die Zulassungskriterien zu erfüllen. Wir haben außerdem eine Liste von möglichen Projekten bereit gehalten, falls Studierende gerne an dem Projekt teilnehmen wollten, aber noch keine eigenen Ideen hatten. Von diesem Angebot habe einige wenige Gruppen Gebrauch gemacht.

Um die Studierenden nicht darauf zu beschränken, ihre Programme am Emulator zu testen, wurde das Board auch für „Live-Experimente“ im Rahmen ohnehin mehrfach pro Woche stattfindender Tutorien zur Verfügung gestellt. In diesen Tutorien standen schon vorher studentische Tutoren dafür bereit, Studierende individuell bei der Arbeit an Übungsaufgaben zu betreuen, so dass der einzige Mehraufwand darin bestand, diese studentischen Tutoren in der Handhabung des LED-Boards zu schulen. Durch dieses Angebot konnte insbesondere sichergestellt werden, dass die Studierenden eine Möglichkeit hatten, sich rechtzeitig mit Performanceunterschieden zwischen ihrem eigenen PC und dem RASPBERRYPI vertraut zu machen und tutorielles Feedback zu ihrem Projekt zu erhalten. Außerdem wurde so eine Umgebung geschaffen in der Studierende sich gegenseitig bei der Arbeit am Projekt beobachten konnten und auf diese Weise ins Gespräch kamen, um wechselseitig weiteres Feedback zum Projekt zu geben und zu erhalten.

## 4 Teilnehmerzahlen und Projekte

Im Wintersemester 2013/2014 nahmen zum Zeitpunkt der Vorstellung des LED-Boards knapp 300 Studierende ernsthaft an der Vorlesung teil, von denen 29 (verteilt auf 18 Gruppen zu einer bis drei Personen) erfolgreich ein Projekt mit dem LED-Board durchführten. Im Sommersemester 2014 führten von etwa gleich vielen Vorlesungsteilnehmern 48 Studierende in 24 Gruppen erfolgreich ein Projekt mit dem LED-Board durch (siehe Tabelle 1). Hinzu kamen in beiden Semestern weitere Anmeldungen, für die bis zum Einsendeschluss jedoch gar keine oder keine zufriedenstellende Einreichung abgegeben wurden. Die

	WS 2013/14	SS 2014
Einzelteilnehmer	10	7
2er-Gruppen	5	10
3er-Gruppen	3	7
Gesamtzahl Teilnehmer	29	48

Tabelle 1: Teilnehmerzahlen in den beiden Projektdurchläufen im Wintersemester 2013/14 und Sommersemester 2014.

geringere Teilnehmerzahl im Wintersemester dürfte auf die spätere Vorstellung des LED-Boards zurückzuführen sein. Das Board befand sich zu dieser Zeit noch in der Entwicklung und vor allem die Java-Bibliothek erhielt in diesem Semester noch mehrere Updates.

Unter den in beiden Semestern eingereichten Projekten fand sich eine sowohl in Art und Idee, als auch programmiererischer Komplexität und Schwierigkeit breite Auswahl an Programmen. Von der nicht-interaktiven Anzeige einfacher Flaggen anlässlich der 2014 stattfindenden Fußball-WM, die mit verschiedenen Animationen ineinander überblendet wurden, über liebevoll animierte Landschaften, Feuerwerke, interaktive Laufschriften (eine davon über Twitter steuerbar!) und Spiele wie *Snake*, *4 gewinnt*, *2048* und *Tetris*, bis zu einem ganzen Programmierungsframework für das Board, dessen Fähigkeiten seine Programmierer durch mehrere verhältnismäßig aufwändige Spiele demonstrierten, war deutlich sichtbar eine weite Spanne von Programmierungsanfängern bis zu bereits erfahrenen Programmierern abgedeckt.

Es kann damit bereits an dieser Stelle festgehalten werden, dass das Ziel, die Kreativität der Studierenden nicht einzuschränken und gleichzeitig sowohl Anfänger als auch Fortgeschrittene anzusprechen, voll erreicht wurde. Im Folgenden werden einige Aspekte zum Umgang mit dem Board allgemein und an exemplarischen Projekten aus beiden Semestern im speziellen diskutiert.

### 4.1 Beispiel 1: Einfache Animation (SS 2014)

Beim ersten hier vorgestellten Projekt handelt es sich um eine einfache Animation. Ein Ballon bewegt sich vor vorbeiziehenden Wolken und einem statischen Hintergrund langsam hin und her (siehe Abbildung 3). Das Projekt erfüllt damit die Minimalanforderungen, ohne in seinem Anspruch an irgendeiner Stelle darüber hinaus zu gehen.

Die Umsetzung erfordert einen sicheren Umgang mit Arrays und Schleifen, ein Verständnis für die korrekte Zeichenreihenfolge bei sich verdeckenden grafischen Objekten und bietet Möglichkeiten, einen strukturierten Programmaufbau zu üben.

Im uns vorliegenden studentischen Quellcode lässt sich tatsächlich eine klare Strukturierung erkennen. Der Code ist vollständig in einer Javaklasse untergebracht und durch Methoden in sinnvolle Einheiten

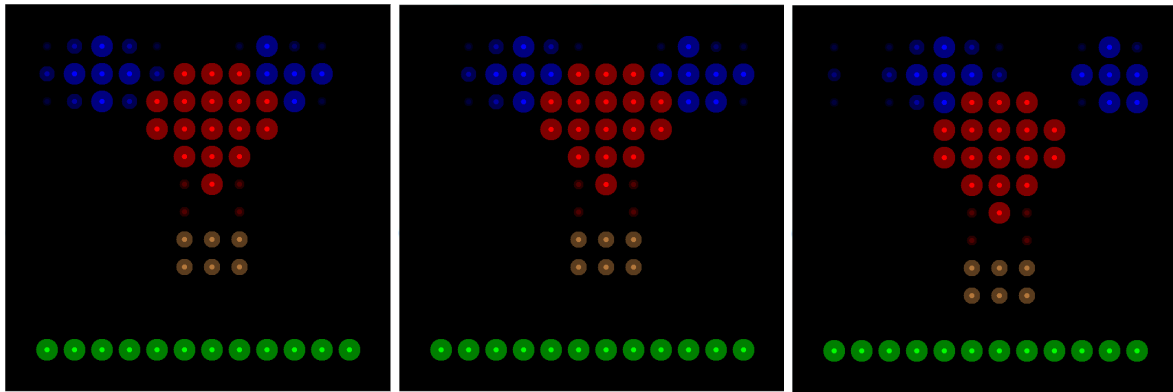


Abbildung 3: Drei direkt aufeinander folgende Screenshots aus einer einfachen Animation mit einem bewegten Ballon vor vorbeiziehenden Wolken, siehe Abschnitt 4.1.

gegliedert. Auch wenn mehr Strukturierung denkbar wäre, ist der verwendete Ansatz für das Problem völlig ausreichend und ein Grundverständnis für die Programmierung deutlich aus dem Quellcode abzulesen. Weitergehende Konzepte der Objektorientierung, die auch Thema der Vorlesung waren, fanden sich in diesem Projekt nicht wieder.

Projektumfang und Programmaufbau legen die Vermutung nahe, dass es sich bei den Studierenden um Anfänger im Bereich der Programmierung gehandelt hat, die mit diesem Projekt Sicherheit erhalten wollten, indem sie in der Vorlesung erworbene grundlegende Programmierfähigkeiten durch weitere Übung festigen wollten. Insbesondere ist nicht zu erkennen, dass die Studierenden Kenntnisse von außerhalb der Vorlesung in das Projekt eingebracht haben oder sich für das Projekt zusätzliche Kenntnisse angeeignet und ausprobiert haben.

#### 4.2 Beispiel 2: Interaktives Spiel (WS 2013/2014)

Das zweite hier vorgestellte Projekt ist eine Implementierung des Spiels „Snake“. Bei diesem Spiel bewegt sich eine vom Spieler zu steuernde Schlange in den vier Grundrichtungen über das Spielfeld und sammelt Äpfel auf, wodurch die Schlange immer länger wird. Das Spiel ist beendet, sobald die Schlange entweder mit sich selbst oder dem Spielfeldrand kollidiert. Tatsächlich war die Spielidee so beliebt, dass allein im Wintersemester 2013/2014 vier verschiedene Teams Implementierungen dieses Spiels eingereicht hatten. Eine dieser Implementierungen ist beispielhaft in Abbildung 4 wiedergegeben.

Für eine gute Implementierung von Snake bietet sich der Einsatz von Objekten und Klassen an, insbesondere die Verwendung einer dynamischen Datenstruktur. Die Schlange besteht aus einer Folge von Einzelpositionen, was die Verwendung eines Arrays oder einer Liste zum Verwalten der Einzelpositionen nahelegt. Eine Fortbewegung der Schlange im Spielverlauf bedeutet aus Spielersicht, dass alle einzelnen Schlangenglieder um ein Feld vorwärts bewegt wer-

den. Aus Sicht des Programmierers ist es effizienter das letzte Schlangenglied zu entfernen und ein neues Glied an den Kopf der Schlange zu setzen. Die mittleren Schlangenglieder müssen dadurch nicht geändert werden. Ist die Schlange einen Apfel, muss ein neues Schlangenglied an die Schlange angehängt werden.

Die hier beschriebenen Mechanismen sind mit einer Liste deutlich einfacher zu implementieren als mit einem Array.

Sowohl Objekte und Klassen, als auch dynamische Datenstrukturen wie Listen wurden in der Vorlesung rechtzeitig eingeführt, um die Verwendung in einem der studentischen Projekte zu ermöglichen. Außerdem erfordert eine Implementierung des Spiels eine einfache Gameloop, d.h. eine Schleife, die wiederholt Nutzereingaben abfragt und die Spielwelt entsprechend anpasst. Zu beachten ist dabei, dass das Spiel beim Warten auf Eingabe nicht pausiert, sondern sich der Spielzustand auch unabhängig von einer Eingabe weiterentwickelt.

Allen vier eingereichten Projekten war anzusehen, dass die Studierenden sich um eine geschickte Strukturierung ihres Codes bemüht hatten, was in einigen Fällen gut, in anderen weniger gut gelang. Die Projekte wurden in zwei bis zehn Klassen untergliedert und in einem Fall auf mehrere Packages verteilt. In einem Projekt erfolgte ein Rückgriff auf die in der Vorlesung vorgestellten Listenkonzepte.

In allen vier Projekten ist es den Studierenden gelungen, ein funktionierendes Snake-Spiel zu implementieren. Neben der Basisfunktionalität fanden sich in allen Projekten zusätzliche Gimmicks wie Eingangs- und Abschlussanimationen, wählbare Schwierigkeitsgrade und Punktzahlanzeigen, was die Motivation der beteiligten Studierenden unterstreicht.

In einem der Projekte ist die selbstständige Verwendung von Threads zu sehen, die anderen drei Projekte verwenden keine über den Inhalt der Vorlesung hinausgehenden Konzepte.



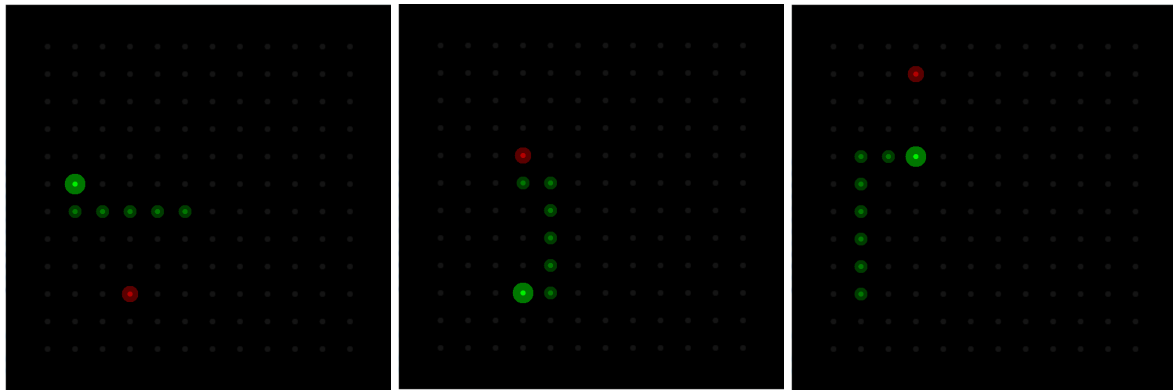


Abbildung 4: Drei Screenshots aus aufeinander folgenden Spielsituationen im Spiel „Snake“, siehe Abschnitt 4.2

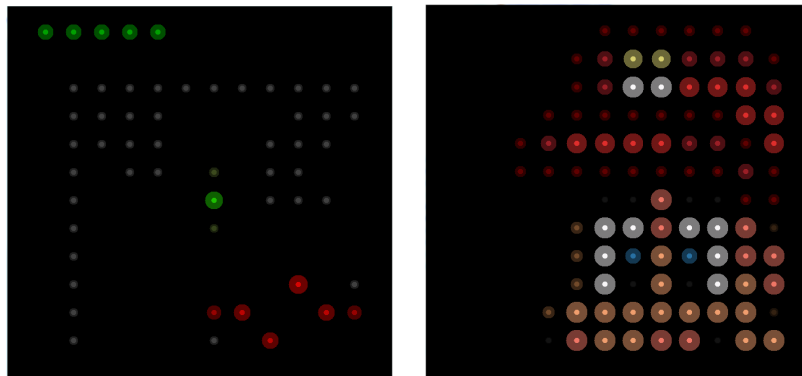


Abbildung 5: Ausgewählte Demonstrationen des studentischen Framework aus Abschnitt 4.3. Links: Ein Helikoptershooter mit dem Spieler in grün, beweglichen Hindernissen in rot und Spielumgebung in grau. Rechts: Supermario, eingelesen aus einer Bilddatei.

### 4.3 Beispiel 3: Framework (WS 2013/2014)

Das dritte hier vorgestellte Projekt hat unsere Erwartungen bei weitem übertroffen. Die dahinter stehenden zwei Studierenden haben sich nicht damit zufrieden gegeben ein Spiel zu schreiben, sondern haben gleich ein ganzes Framework für das LED-Board angefertigt und dieses Framework an mehreren nicht trivialen Anwendungen und Spielen demonstriert. In Abbildung 5 sind zwei ausgewählte Beispiele zu sehen. Im Framework ist unter anderem Unterstützung für das Verarbeiten von Polygonen und Sprites zu finden. Des weiteren findet sich eine Kollisionserkennung für diese Objekte, wahlweise per Hitbox oder pixelbasiert. Außerdem besteht die Möglichkeit, Bilder aus Bitmapen oder gespeicherten Dateien in üblichen Grafikformaten auf dem Board darzustellen. Dabei unterstützt das Framework auch das Scrolling von Bildern oder Masken, deren Auflösung größer als die des LED-Boards ist. Zur Demonstration des Frameworks waren mehrere Programme beigelegt, die das Framework nutzen und sehr gut strukturiert sind. Framework und Demonstrationsprogramme bestehen aus insgesamt 95 Klassen, die in 26 Packages organisiert sind.

Der hohe Organisationsgrad, die Komplexität und die Verwendung fortgeschrittener Konzepte wie

Quadtrees zeigen, dass die beiden Studierenden schon vor unserer Programmierungsvorlesung umfangreiche praktische Erfahrungen gesammelt haben.

Der Aufwand für dieses Framework muss schon allein aufgrund des Umfangs sehr hoch gewesen sein und belegt, dass mit dem LED-Board auch fortgeschrittene Studierende angesprochen und motiviert werden können.

### 4.4 Allgemeines zu den Projekten

Die vorgestellten Projekte gewähren einen guten Eindruck von der Bandbreite der abgegebenen Projekte. Unter allen angenommenen und am Ende in der Vorlesung präsentierten Programmen fanden sich naturgemäß auch ein paar schwächere Projekte, deren Urheber vor allem zwecks Erwerb der Klausurzulassung teilnahmen. Der Anteil dieser Projekte an sämtlichen präsentierten Projekten blieb allerdings mit unter einem Fünftel erfreulich gering. Insgesamt ungefähr ein Viertel der Anmeldungen führte allerdings gar nicht erst zu Abgaben.

Aus den Erfahrungen der Tutorien, des Codereviews und des uns vorliegenden Quellcodes offenbarte sich an vielen Stellen eine große Experimentierfreude. So versuchten sich viele Studierende in der Verwendung von Konzepten, die bis dahin nicht Teil ihres Studi-

Projekt	LOC	Klassen
Einfache Animation <small>(SS 2014)</small>	118	1
Interaktives Spiel <small>((WS 2013/14)</small>	302	3
Framework <small>((WS 2013/14)</small>	4509	95
Durchschnitt aller Projekte	710	6

Tabelle 2: Umfang ausgewählter Projekte und Durchschnitt aller Projekte. Die LOC-Metrik berücksichtigt nur reine Codezeilen und ignoriert Leerzeilen und Kommentarzeilen.

ums waren. Vielfach wurden Threads verwendet, die JavaAPI wurde mehrfach für die Generierung von Zufallszahlen und für Filehandling verwendet und damit einhergehend war auch Exceptionhandling zu sehen. Ferner zeigten sich immer wieder Studierende offen für Vorschläge, die über ihr ursprüngliches Projektziel hinausgingen. Nicht immer waren diese Experimente von Erfolg gekrönt, dennoch waren sie ein weiteres Indiz für die Motivation. Umgekehrt scheiterten leider auch einige Studierende an ihren ursprünglichen Zielen und mussten sie herunterschrauben. In den Codereviews zeigte sich zudem, dass auch in den Teams stets alle Beteiligten tatsächlich programmiert hatten. Es liegt nahe, dass dies zumindest zum Teil auch darauf zurückzuführen ist, dass das Board keine Möglichkeiten bietet, sich mit anderen Arbeiten (z.B. dem Erstellen hübscher Icons) einzubringen.

Allen Projekten war zu eigen, dass den Teilnehmern die Notwendigkeit guter Codestrukturierung unmittelbar vor Augen geführt wurde, wenn auch nicht alle Studierenden in der Lage waren, eine solche Strukturierung mit ihren Mitteln zu erreichen. Einige Daten zum Umfang der oben vorgestellten Projekte gemessen in Codezeilen (LOC) sowie zum Organisationsgrad gemessen in der Zahl der Klassen sind in Tabelle 2 aufgeführt. Der in der Tabelle angegebene Durchschnitt bezieht sich auf alle abgegebenen Projekte aus beiden Semestern. Dabei ist zu bemerken, dass sehr starke Unterschiede zwischen einzelnen Projekten auftraten: Drei Projekte mit je über 1000 LOC waren jeweils in nur einer Klasse realisiert, während fünf andere Projekte jeweils einen Schnitt von weniger als 50 LOC pro Klasse erreichten.

In den Tutorien und spätestens im abschließenden Codereview konnten den Studierenden in den meisten Fällen zahlreiche Hinweise gegeben werden, die ihren Code verbessern und vor allem den Sinn vieler in der Vorlesung vorgestellten Konzepte an interessanten, da von den Studenten selbst ausgedachten und gewählten, Szenarien verdeutlichen. Softwarekonstruktion und wie man guten Code schreibt waren nur am Rande Thema unserer Programmierungsvorlesung. Durch das LED-Projekt konnten die meisten Teilnehmer dennoch in einem Maß für diese Problematiken sensibilisiert werden, das wir ohne das Projekt nicht erreicht hätten.

Testatpunkte	Spaß	Training	Preis
82%	66%	48%	21%

Tabelle 3: Motivation zur Teilnahme (Mehrfachnennungen möglich)

## 5 Rückmeldung der Studierenden

Neben persönlichem Feedback haben wir zum Abschluss der Projekte in beiden Semestern jeweils einen anonymen Fragebogen zum LED-Board ausfüllen lassen. Gefragt wurden ausschließlich Studierende, die auch ihr Projekt vorgestellt hatten. Die hier präsentierten Ergebnisse wurden aus insgesamt 71 ausgefüllten Fragebögen beider Semester zusammengetragen. Gefragt wurde

- nach den Gründen zur Teilnahme an dem LED-Projekt,
- ob sich das eigene Programmierverständnis nach eigener Einschätzung verbessert hat,
- ob die Teilnehmer für sich ein überdurchschnittliches Klausurergebnis erwarteten,
- wie sich die Motivation der Teilnehmer mit dem Projektverlauf entwickelt hat
- und ob das Projekt die Erwartungen der Teilnehmer erfüllt hat.

Außerdem bot der Fragebogen Raum für freie Kommentare aller Art.

### 5.1 Gründe zur Projektteilnahme

Als mögliche Gründe für die Projektteilnahme waren

- Testatpunkte
- Spaß
- Training der eigenen Fähigkeiten
- der Sachpreis für den ersten Platz
- und sonstiges

gegeben, wobei Mehrfachnennungen möglich waren. Das zusammengefasste Ergebnis ist in Tabelle 3 angegeben. Insgesamt haben 82% der Teilnehmer den Erwerb von Testatpunkten als Grund genannt. Bemerkenswert ist dabei jedoch, dass viele Teilnehmer zum Zeitpunkt der Vorstellung ihrer Projekte ihre Klausurzulassung bereits hatten, und sie dennoch zu Vorstellung und Codereview gekommen sind.

Sehr erfreulich ist die hohe Zahl der Teilnehmer die Spaß am Board (66%) oder Training der eigenen Fähigkeiten (48%) als Motivation angegeben haben. Zu untersuchen bleibt, aus welchen Gründen sich Studierende gegen eine Teilnahme am LED-Projekt entschlossen haben.

hoch	fallend	niedrig	steigend
46%	21%	6%	27%

Tabelle 4: Motivationsverlauf

## 5.2 Verbesserung des eigenen Programmierverständnisses

Die Verbesserung ihres eigenen Programmierverständnisses sollten die Teilnehmer auf einer Skala von 0 (gar nicht) bis 4 (sehr stark) einschätzen. Über alle Teilnehmer gemittelt hat sich hier ein Wert von ca. 2,3 ergeben. Für den Großteil der Teilnehmer erbrachte das Projekt im subjektiven Eindruck also eine spürbare Verbesserung der eigenen Fähigkeiten.

Studierende, die bei der Motivation angegeben haben, ihre eigenen Fähigkeiten trainieren zu wollen, haben in der Kategorie „Verbesserung des eigenen Programmierverständnisses“ im Schnitt ca. 2,6 angegeben, was signifikant über dem Gesamtschnitt von 2,3 liegt.

## 5.3 Erwartungen an die Klausur

Mit den Antwortmöglichkeiten „Ja“ und „Nein“ sollten die Teilnehmer einschätzen, ob sie ein überdurchschnittliches Klausurergebnis erzielen werden. 45% erwarteten ein überdurchschnittliches Ergebnis, die restlichen 55% nicht. Da die Umfrage anonym durchgeführt wurde, kann leider nicht überprüft werden, ob die Erwartungen mit den tatsächlichen Klausurergebnissen übereinstimmen.

## 5.4 Motivationsverlauf

In der Frage nach dem Motivationsverlauf war eine der vier Möglichkeiten „hoch“, „fallend“, „niedrig“, „steigend“ zu wählen. Die Ergebnisse sind in Tabelle 4 aufgeführt.

73% der Studierenden gab einen dauerhaft hohen oder mit der Zeit steigenden Motivationsverlauf an. Interessant zu erfahren wäre für zukünftige Erhebungen, warum bei den übrigen 27% die Motivation gelitten hat. Es ist nicht ersichtlich, ob die Gründe dafür im LED-Projekt selbst oder in äußeren Umständen wie z.B. Zeitdruck liegen.

Eine weitere Analyse der Daten zeigt, dass Teilnehmer die aufgrund von Spaß am Board oder um die eigenen Fähigkeiten zu trainieren teilnahmen, leicht überdurchschnittlich oft eine hohe oder steigende Motivation aufwiesen. Der Preis hatte dagegen keinen erkennbaren Einfluss auf die Motivation. Teilnehmer die Testpunkte als einen von mehreren Teilnahmegründen angegeben haben, wiesen im Schnitt einen geringfügig schlechteren Motivationsverlauf auf. Betrachtet man die Teilnehmer, die ausschließlich der Testpunkte wegen ein Projekt eingereicht hatten, weist sogar nur ein Drittel dieser Teilnehmer eine hohe oder steigende Motivation auf.

## 5.5 Erwartungen an das Projekt

Auf einer Skala von 0 (gar nicht) bis 4 (vollständig) sollte angegeben werden, ob die Erwartungen der Teilnehmer an das Projekt erfüllt wurden. Mit dem Mittelwert von ca. 3,0 können wir erfreulicherweise feststellen, den Erwartungen im Wesentlichen gerecht geworden zu sein.

## 5.6 Freie Kommentare

Neben der Umfrage sprechen vor allem die Einzelkommentare, die direkten Rückmeldungen an die Projektbetreuer sowie die rege Teilnahme an dem Wettbewerb für eine große Akzeptanz des Projektes. Kritisiert wurden von den Studierenden insbesondere technische Aspekte wie z.B. die geringe Zahl von 12x12 LEDs und die Leistungsfähigkeit des RASPBERRYPI3.

Die Umfrageauswertung lässt insgesamt den Schluss zu, dass das Board und der zugehörige Emulator gute Werkzeuge zum selbstständigen Training sind und von den Studierenden auch zu diesem Zweck akzeptiert werden.

## 6 Ausblick und weiterer Einsatz

In kommenden Programmierveranstaltungen werden wir sicherlich wieder einen Wettbewerb auf dem LED-Board anbieten. Da das LED-Board fertiggestellt ist und die Bibliothek von ihren Kinderkrankheiten befreit, ist der durch den Wettbewerb entstehende Mehraufwand für das Lehrpersonal sehr gering für den deutlich positiven Effekt.

Neben der gegenwärtigen Verwendung ist es auch denkbar, das Board oder zumindest den Emulator in die eigentliche Vorlesung und/oder die Übungen zu integrieren. Insbesondere Schleifen und Arrays lassen sich an dem Board sehr anschaulich demonstrieren. Für Übungsaufgaben bieten sich Board und Emulator ebenso an.

Mit kleinen Anpassungen der LED-Bibliothek ist auch eine Einbindung in unserer automatisches Übungs- und Prüfungssystem JACK (Striwe u. a., 2009) möglich. Sich auf automatisch auswertbare Aufgaben zu konzentrieren würde allerdings einen sehr großen Aufwand und eine starke Einschränkung der Nutzungsmöglichkeiten des Boards bedeuten.

Neben dem Einsatz in der Erstsemestervorlesung wurde das LED-Board auch schon mit interessierten Schülern in Workshops, fortlaufenden AGs oder anderen Veranstaltungen zur Studiengangswerbung in der Informatik verwendet.

## 7 Fazit

Insgesamt betrachten wir den Einsatz des LED-Boards als erfolgreich, da (1) das Board von den Studierenden wie erwartet einfach und ohne nennenswerte technische Probleme nutzbar war, (2) sowohl einfache als auch anspruchsvolle Projekte auf dem Board realisiert wurden, (3) optisch ansprechende Ergebnisse in einer wettbewerbsartigen Form im Hörsaal



präsentiert werden konnten, (4) eine große Breite an verschiedenartigen Projekten realisiert wurde und (5) die Mehrheit der teilnehmenden Studierenden eine hohe Motivation aufwies und durch ihre Arbeit belegte.

Dem Aufwand bei der Vorbereitung des Boards steht somit insbesondere ein messbarer Erfolg bei der Steigerung der Motivation der Studierenden entgegen, der mindestens mit einem subjektiv höheren Lernerfolg verbunden ist.

## Literatur

[Henriksen u. Kölling 2004] HENRIKSEN, Poul ; KÖLLING, Michael: greenfoot: Combining Object Visualisation with Interaction. In: *OOPSLA '04: Companion to the 19<sup>th</sup> annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*. New York, NY, USA : ACM, 2004. – ISBN 1-58113-833-4, S. 73–82

[Striwe u. a. 2009] STRIEWE, Michael ; BALZ, Moritz ; GOEDICKE, Michael: A Flexible and Modular Software Architecture for Computer Aided Assessments and Automated Marking. In: *Proceedings of the First International Conference on Computer Supported Education (CSEDU), 23 - 26 March 2009, Lisboa, Portugal* Bd. 2 INSTICC, 2009, S. 54–61